Удобно е да гледаме точките в реда на намаляване на y координатата и да ги запазваме в някаква структура от данни. По този начин, докато разглеждаме някоя точка, няма да се налага да отсяваме точките по-долу (те нямат никакво значение за нас) и точките по-горе (те ни интересуват), защото точките по-долу все още няма да са влезли в нашата структура от данни.

Нека помислим какви функции трябва да изпълнява нашата структура от данни. За някоя точка ще искаме да намерим първата отляво и първата отдясно, които имат температура на топене по-голяма или равна на тази на текущата точка. Разбира се, тази структура ще трябва да поддържа добавяне на нови точки. Оказва се, че добра работа върши интервалното дърво. В него ще поддържаме максималната температура на топене на някоя точка в даден интервал от абсцисната ос.

Нека разглеждаме намирането на първата точка отляво (за точката отдясно ще бъде аналогично). Един възможен подход е да търсим двоично тази точка. При двоичното търсене на всяка стъпка фиксираме интервала с десен край x координатата на текущата точка, за който ще трябва да проверим в дървото дали съдържа точка с по-висока температура. Ако такава точка не се съдържа, удължаваме интервала наляво. В противен случай го скъсяваме отляво. Накрая ще намерим последния интервал, който не съдържа такава точка и съответно ще знаем колко може да е дебела пръчката отляво. Сложността на това търсене обаче е log2MAX\_C, където MAX\_C е максималната абсциса. Това не ни устройва.

Ще оптимизираме процеса, като извършваме търсенето, докато се движим по дървото. Нека се намираме в произволен интервал. Първо пробваме да намерим търсената точка в дясната част на интервала надолу по дървото. Ако намерим точка, я връщаме. Ако не намерим, търсим точка в лявата част на интервала. Тук за да запазим сложност на търсенето logMAX\_C, трябва да спрем търсенето, ако се окажем в интервал, за който сме сигурни, че не съдържа желаната точка, т.е. интервал, който е строго надясно от абсцисата на фиксираната точка или изцяло наляво от абсцисата, но максималната температура е по-малка от температурата на фиксираната точка. С това съображение си осигуряваме, че за едно търсене ще стигнем до най-много едно листо. Добавянето на нова точка е стандартно и не би трябвало да затрудни опитните състезатели.

Общата сложност на алгоритъма е O(NlogN + NlogMAX\_C); първото събираемо идва от сортирането, а второто – от заявките в дървото за всяка от N-те точки.